Research Article

Lightweight Ship Detection Methods Based on YOLOv3 and DenseNet

Zhelin Li, Lining Zhao, Xu Han, and Mingyang Pan

Dalian Maritime University, Dalian 116026, China

Correspondence should be addressed to Mingyang Pan; panmingyang@dlmu.edu.cn

Received 20 April 2020; Revised 10 September 2020; Accepted 11 September 2020; Published 28 September 2020

Academic Editor: Feng-Jang Hwang

Copyright © 2020 Zhelin Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ship detection is one of the most important research contents of ship intelligent navigation and monitoring. As a supplement to classical navigational equipment such as radar and the Automatic Identification System (AIS), target detection based on computer vision and deep learning has become a new important method. A target detector called YOLOv3 has the advantages of detection speed and accuracy and meets the real-time requirements for ship detection. However, YOLOv3 has a large number of backbone network parameters and requires high hardware performance, which is not conducive to the popularization of applications. On the basis of YOLOv3, this paper proposes a lightweight ship detection model (LSDM) in which the backbone network is improved by using dense connection inspired from DenseNet, and the feature pyramid networks are improved by using spatial separation convolution to replace normal convolution. The two improvements reduce parameters and optimize the network structure greatly. The experimental results show that, with only one-third of parameters of YOLOv3, the LSDM has higher accuracy and speed for ship detection. In addition, the LSDM is simplified further by reducing the number of densely connected units to form a model called LSDM-tiny. The experimental results show that, LSDM-tiny has similar detection speed with YOLOv3-tiny, but has a lot higher accuracy.

1. Introduction

In the recent years, object detection technologies based on deep learning have received more and more attention in the areas of ship intelligent navigation and ship monitoring [1, 2]. The rapid perception of the navigational environment is the prerequisite for ships to sail safely or for shore base to monitor ships real-time [3]. The perception data collected by radar and the AIS play an important role in these application areas [4]. However, how to detect the ship targets by cameras visually and automatically has become a new challenge for realizing autonomous navigation or intelligent supervision.

Normally, there are two main types of object detection algorithms based on deep learning. One is a two-stage detection algorithm, such as Fast R-CNN [5] and its improved versions Faster R-CNN [6] and Mask R-CNN [7]. Two-stage algorithms first select candidate targets through the regional proposal networks (RPN) and, then, complete the prediction of the target location and category through the detection

network. The other type is the one-stage detection algorithm, such as SSD [8], YOLO [9], and RetinaNet [10]. One-stage algorithm eliminates the step of region proposal and directly returns the target position and category. All detection algorithms are generally trained on large public data sets to pursue high accuracy; however, they also require high hardware performance to be trained and executed.

Unlike ordinary detection models, lightweight detection models with smaller parameters aim to run on mobile devices or computers with weak computing capabilities. Iandola et al. proposed a lightweight model called SqueezeNet, in which the parameters are compressed to one-fifth of AlexNet by using a small convolution kernel and reducing the number of input and output channels of the convolution layer [11]. Huang et al. proposed a densely connected network called DenseNet, in which feature reuse was used to reduce the parameter of deep networks [12]. Howard et al. built a lightweight deep neural network called MobileNets through deep separable convolutions [13]. Zhang et al.

proposed an efficient convolutional neural network called ShuffleNet, in which the computational cost was greatly reduced through pointwise group convolution and channel shuffling while ensuring accuracy [14].

For ship detection, in addition to accuracy, improving the detection speed is also important to adapt the model to existing hardware condition. As the YOLOv3 has a relatively balanced performance in detection accuracy and speed [15] and the DenseNet has obvious effect on reducing parameters, this paper focused on researching lightweight ship detection models by combining the YOLOv3 and DenseNet and provided a new lightweight detector for high-accuracy ship detection. The contribution of this paper includes the following:

- (1) We propose a lightweight ship detection model called the LSDM, with one-third parameters of the YOLOv3 network, and higher average accuracy of 94% for ship detection
- (2) We propose a simpler version of the LSDM called LSDM-tiny, with one-eighth parameters of the YOLOv3 network, double detection speed, and average accuracy of 93.5% for ship detection

The rest of the paper is organized as follows. Section 2 outlines the related works of ship detection, Section 3 introduces the network details of the LSDM and LSDM-tiny based on YOLOv3 and DenseNet. Section 4 gives the experimental results and comparative analysis. Finally, Section 5 gives the conclusions and future work.

2. Related Works

For image detection, many studies have made improvements on the basic detection model. Fang et al. used DCGAN to generate samples and training in the image recognition model which based by the CNN to improve the accuracy of image recognition [16]. Meng et al. proposed an approach based on the Faster R-CNN which corresponds multiple steganographic algorithms to complex texture objects was presented for hiding secret messages [17]. Cui et al. proposed an effective approach to automatically identify photographic images and computer-generated graphics based on deep convolutional neural networks (DCNNs) by deepening the network structure [18].

For ship detection, there are two types of images to be utilized, radar image and visible image. Generally, the radar image covers a wider range, and the visible image provides more detailed information. Dong et al. improved the R-CNN and proposed a multiangle box-based rotation insensitive object detection structure for detecting VHR (Very-High-Resolution) ship images [19]. Yang et al. proposed a detection model based on the multitask rotational region convolutional neural network to detect dense ships and predict the direction of ship navigation [20]. Fan et al. proposed a ship detection method for PolSAR (Polarization Synthetic Aperture Radar) images based on modified the Faster R-CNN which is still difficult to detect small targets near the shore [21]. Zhang et al. optimized the backbone network of the Faster R-CNN by using SVM to divide the

area of interest and improved effectively the ship detection effect from SAR images [22]. Jiao et al. proposed a Faster R-CNN detection framework based on densely connected multiscale neural networks for detecting SAR ship targets in multiscale and multiscenarios [23]. Kim et al. proposed a hybrid method in which Faster R-CNN was used to detect ships in each image, and the detected ships are, then, gathered over time to compute probabilities for Bayesian fusion to determine the classification of ships [24]. The abovementioned methods are two-stage algorithms which have effectively improved detection effect, but need a long detection time to extract proposal regions through the selective search algorithm or RPN, so they are difficult to implement real-time detection.

There are also many ship detection methods based on one-stage detection algorithms. An et al. proposed an improved RBox-based target detection framework to improve detection accuracy and recall [25]. Liu et al. proposed an improved YOLOv3 algorithm based on the Darknet to realize the detection and tracking of ships in monitored water areas [26]. Zhang et al. proposed a ship target tracking algorithm based on the YOLO method in which the characteristics of HOG and LBP were combined to solve the problem of missing or inaccurate positioning [27]. Wang et al. used SSD to perform transfer learning on Sentinel-1 SAR ship images, improving the detection accuracy and overall performance [28]. Wang et al. used RetinaNet to detect multiscale Gaofen-3 ship images and obtained higher detection accuracy [29]. Chang et al. proposed a YOLOv2 detection structure with reduced number of layers for detecting SAR ship images, which accelerated the inference process while maintaining similar detection effects [30]. Zhang and Zhang drew on YOLO's regression ideas and proposed a detection structure composed of a backbone convolutional network and a detection convolutional neural network, which has a faster detection speed [31]. Generally, one-stage algorithms have higher detection speed but worse accuracy than two-stage algorithms. The abovementioned methods mainly improved the detection accuracy based on one-stage detection algorithms. However, their backbone networks still have a large number of parameters, which not only bring the risk of overfitting and a big model size but also constrain them to archive higher detection speed.

To implement real-time ship detection, Qi et al. proposed an improved Faster R-CNN algorithm by scene reduction technology to reduce the target scale during searching [32]. Lin et al. proposed a new network for ship detection in SAR Images based on the Faster R-CNN, which improved the detection performance and execution speed through squeeze and excitation mechanisms [33]. Zhang et al. proposed a ship detection model called CCNet which uses a cascaded CNN model with REM and RDM, which has five times less computation than other algorithms [34]. The abovementioned methods tried to improve the detection performance and speed based on two-stage algorithms by reducing the weight of the object detection proposal region. However, compared to one-stage end-to-end models, the candidate areas selection step of the methods is still an unignorable factor which would slow down the detection

speed obviously. Zhang et al. proposed a high-speed SAR ship detection model inspired from the experiments of MobileNet, YOLO, SSD, and DenseNet, in which deep separable convolutions instead of ordinary convolutions were used to reduce parameters and improve detection speed significantly; however, it leads to part loss of accuracy [35].

So, with review and analysis of the related works of ship detection, this paper aims to study lightweight ship detection models base on one-stage algorithm, which would keep the accuracy as is possible while reducing the number of parameters and increasing the detection speed.

3. Lightweight Ship Detection Methods

3.1. YOLOv3. YOLOv3 is an end-to-end object detection model, and its network structure includes a backbone network and a detection network [15], as shown in Figure 1. In the backbone network, an input image is scaled to the size of 416×416 without changing the aspect ratio, and down sampled 5 times to extract feature maps. Then, the feature maps with the size of 13×13 , 26×26 , 52×52 are output, respectively, to three branches of the detection network to form a feature pyramid structure, in which the feature maps in a lower branch are concatenated with the ones in its next branch by up sampling. Finally, the outputs of the feature pyramid network are sent to a regression section to carry out bounding box and category prediction.

The original backbone network of YOLOv3 is Darknet-53. Darknet-53 includes 52 fully convolution layers, in which 46 layers are divided into 23 residual units with 5 different sizes [15]. The residual units are designed to avoid the vanishing-gradient problem inspired from the Resnet [36]. The Darknet-53 is a complex network, and its 40549216 parameters provide a guarantee for the detection accuracy. However, for the object detection with single category such as ship, the excessively huge parameters would bring overfitting risk and slow down the detection speed.

YOLOv3-tiny is a simplified version of YOLOv3, its backbone network only includes 7 convolutional layers and 6 max-pooling layers, and its feature pyramid network is also simplified by removing the maximum-scale prediction branch and reducing the number of convolutional layers in the other two branches. So, YOLOv3-tiny has faster detection speed than YOLOv3 due to its shallow and simple network structure; however, its detection accuracy is lower obviously than YOLOv3.

Therefore, for fast ship detection, it is important to maintain the depth of network for capturing enough features to ensure detection accuracy while reducing network parameters to speed up. In addition, the ship objects are relatively small in the images, when detecting them by Darknet-53, and their shallow features should be clear but their deep features are easy lost after multiple down sampling. So, how to utilize the shallow features as much as possible to improve the detection accuracy becomes the key issue should be solved. This paper proposes a method of feature reusing inspired from DenseNet to achieve this goal.

3.2. Densely Connected Unit. Different with the ResNet, DenseNet solves the vanishing-gradient problem by connecting each layer to every other layer in a feed-forward fashion. As shown in Figure 2, DenseNet is a narrow network, in which each layer accepts inputs from all previous layers and passes the feature maps to all subsequent layers [12]. Each layer reuses the global features and adds only a few new features remaining other features unchanged. This mechanism of feature reuse makes the DenseNet has fewer parameters than traditional convolutional neural networks. In addition, as each layer can access directly the gradient from the loss function, the DenseNet optimizes the information flow and gradient throughout network, which make it easy to be trained and has low overfitting risk on small training data set.

From Figure 2, it can be seen that DenseNet should keep the size of feature maps in all layers consistent to connect each other. However, as we all know, for every deep convolutional neural network, the feature maps must be downsampled multiple times to expand the receptive field gradually and improve computational efficiency. To solve this contradiction, DenseNet divides the network into several densely connected units, and the convolution layers in a densely connected unit keep same size of feature map, while a densely connected unit downsamples to its next unit by average pooling.

As shown in Figure 3, there are two convolution layers in a densely connected unit. The first convolution layer called the bottleneck layer is used to control the number of feature maps. The second convolution layer is used for feature extraction, and the number of convolution kernels in this layer is called growth rate, as it represents the number of new features added into the global features in this layer. Since a densely connected unit only contains new features and reuses the global features input from all previous layers, the number of convolution kernels is much less than that of the ordinary convolution layer. So, the parameters of DenseNet is reduced greatly; in addition, the feature reuse method is also useful to improve its detection accuracy for small object by keeping shallow features in the final global features.

3.3. Backbone Network of the LSDM. For developing a lightweight ship detection model (LSDM), a new backbone network is constructed based on the combination of darknet-53 and DenseNet. Table 1 shows structure of the LSDM backbone network:

In the backbone network, the densely connected unit of DenseNet is used to replace the residual units in DarkNet-53. Within a densely connected unit, the number of convolution kernels in the bottleneck layer is set to 128, and the growth rate of the feature extraction layer is set to 32. That is, for each densely connected unit, the input feature maps will be compressed, firstly, to 128, and then, 32 new feature maps will be added to the global feature. The size of feature maps become smaller as the layer goes deeper, and more feature maps are needed to keep the feature semantic information abundant. That is, as the size of feature maps decreases, more densely connected units are needed to increase their number. Therefore, in the whole backbone network, 5 levels

FIGURE 1: The structure of YOLOv3.

FIGURE 2: The structure of dense connection.

of densely connected units are adopted with increased numbers of 1, 2, 4, 8, and 16, and the average pooling is used to downsample from one level to its next level. As a result, the backbone network contains 63 convolution layers, and the final number of global feature maps is 1024.

Although there are 11 more layers than Darknet-53, due to the existence of bottleneck layers in densely connected units and feature reuse mechanism, the proposed backbone has fewer parameters than Darknet-53. The parameter number of a convolution layer can be calculated by the following equation:

$$P = k_h \times k_w \times C_{in} \times C_{out}, \quad (1)$$

where P is the number of parameters, $k_h \times k_w$ is the size of the kernels, C_{in} is the number of input channels, and C_{out} is the number of output channels.

The parameters of the last residual unit in Darknet-53 and the last densely connected unit in the LSDM backbone network are shown in Table 2. In terms of the whole network, the parameter number of the LSDM backbone network is 3175264, which is just 7.8% of 40549216 that of the Darknet-53.

FIGURE 3: The structure of the densely connected unit.

TABLE 1: The structure of the LSDM backbone network.

	Type	Filters	Size/stride	Output size
	Conv	32	3*3/1	416*416
	Avg_pool		2/2	208*208
1x	Conv	128	1*1/1	208*208
	Conv	32	3*3/1	208*208
	Avg_pool		2/2	104*104
2x	Conv	128	1*1/1	104*104
	Conv	32	3*3/1	104*104
	Avg_pool		2/2	52*52
4x	Conv	128	1*1/1	52*52
	Conv	32	3*3/1	52*52
	Avg_pool		2/2	26*26
8x	Conv	128	1*1/1	26*26
	Conv	32	3*3/1	26*26
	Avg_pool		2/2	13*13
16x	Conv	128	1*1/1	13*13
	Conv	32	3*3/1	13*13

*The units with gray color are densely connected units (its structure is shown in Figure 3).

TABLE 2: Parameters comparison between the residual unit and densely connected unit.

Structure	Layer	Kernel size	Input channel	Output channel	Parameters
Residual unit	Conv	1*1	1024	512	524288
	Conv	3*3	512	1024	4718592
Densely connected unit	Conv	1*1	992	128	126976
	Conv	3*3	128	32	36864

3.4. Backbone Network of LSDM-Tiny. A further compressed backbone network for LSDM-tiny is also investigated. Since the number of global feature maps is related to the number of densely connected units, reducing the number of densely connected units will further decrease the parameters; however, it will also decrease the detection accuracy. In order to keep accuracy as much as possible while reducing densely connected units, a compromise method is applied in the backbone network of LSDM-tiny. There are only two densely connected units no matter which levels; however, a convolution layer with a convolution kernel size of 1*1 is added between level 2 and level 3 and between level 3 and level 4, respectively, to increase the dimension of features. The whole structure of the backbone

network of LSDM-tiny is shown in Table 3, and its parameter number is 1291104, which is only 40% of the backbone network of the LSDM and 3% of the Darknet-53.

3.5. LSDM and LSDM-Tiny. The abovementioned backbone networks are, then, used to replace the Darknet-53 on the basis of YOLOv3 to form completed ship detection network LSDM and LSDM-tiny. The overall structure of the LSDM is shown in Figure 4. The output of densely connected units in the last three levels in the backbone network are sent to the right feature pyramid network as inputs. Finally, the three-scale feature maps output by the feature pyramid are used to carry out detection.

TABLE 3: The structure of the LSDM-tiny backbone network.

	Type	Filters	Size/stride	Output size
1x	Conv	32	3*3/1	416*416
	Conv	128	1*1/1	416*416
	Conv	32	3*3/1	416*416
	Avg_pool		2/2	208*208
2x	Conv	128	1*1/1	208*208
	Conv	32	3*3/1	208*208
	Conv	256	1*1/1	208*208
	Avg_pool		2/2	104*104
2x	Conv	128	1*1/1	104*104
	Conv	32	3*3/1	104*104
	Conv	640	1*1/1	104*104
	Avg_pool		2/2	52*52
2x	Conv	128	1*1/1	52*52
	Conv	32	3*3/1	52*52
	Avg_pool		2/2	26*26
	Conv	128	1*1/1	26*26
2x	Conv	32	3*3/1	26*26
	Avg_pool		2/2	13*13
	Conv	128	1*1/1	13*13
	Conv	32	3*3/1	13*13

*The units with gray color are densely connected units (their structure is shown in Figure 3).

*The dense_unit is the abbreviation of the densely connected unit

FIGURE 4: The overall structure of the LSDM network.

In addition to the backbone network, the feature pyramid network is also improved. The 3*3 convolution layer of the last branch is split into a combination of 3*1 and 1*3 convolution layers which can be called spatial separation

convolution. Spatial separation convolution can reduce the parameter number by half, and it will perform best in the last branch because there are mostly small-scale feature maps. Table 4 shows the parameters comparison between the

TABLE 4: Parameters comparison of standard convolution and spatial separation convolution.

Structure	Layer	Kernel size	Input channel	Output channel	Parameters
Conv2d	Conv	3*3	512	1024	4718592
Separable conv	Conv	3*1	512	512	786432
	Conv	1*3	512	1024	1572864

FIGURE 5: Samples of the training dataset.

TABLE 5: Comparison of model training results.

	Recall	Precision	AP	F1	Parameters (M)	FPS
LSDM	0.954	0.851	0.940	0.899	20.02	16.97
YOLOv3	0.946	0.830	0.930	0.884	61.47	13.48
LSDM-tiny	0.958	0.804	0.935	0.874	7.05	27.09
YOLOv3-tiny	0.916	0.707	0.881	0.798	8.66	29.49

standard convolution and spatial separation convolution in the feature pyramid network.

In summary, by using new backbone network and spatial separation convolution, the total parameters of the LSDM is 20022112, which is about 32% of YOLOv3.

LSDM-tiny can be transformed from the LSDM by replacing the backbone network with that for LSDM-tiny (As shown in Table 3) and deleting the branch on the 52*52 scale in the feature pyramid network. The total parameters of LSDM-tiny are 7054368, which is about 36% of the LSDM and about 12% of YOLOv3.

3.6. Model Training Details. In order to improve the detection accuracy of the LSDM and LSDM-tiny and speed up their training speed, some tricks also can be added. Firstly, the LeakyReLU activation function is used to replace ReLU (Rectified Linear Units), and the negative half-axis slope of the function is set to 0.1; its formula is shown in the following equation:

$$f(x_i) = \begin{cases} x_i, & x_i \geq 0, \\ \frac{x_i}{a}, & x_i < 0, a = 0.1. \end{cases} \quad (2)$$

Secondly, to make the models converge faster, momentum is added to the SGD optimizer, and the improved optimization function is shown in the following equation:

$$\begin{aligned} v &= \beta \times v - \alpha \times dx, \\ x &= x + v, \end{aligned} \quad (3)$$

where β is the momentum parameter set to 0.949, and the initial value of variable v is set to 0.

Thirdly, in order to reduce the risk of overfitting, weight decay for the parameters of the convolution layers is used, and its attenuation coefficient is set to 0.000489.

4. Experiments and Analysis

4.1. Ship Image Dataset. The ship image dataset contains 2270 pictures captured by web crawlers and processed by data enhancement methods such as random image flipping, noise addition, and color enhancement. Also, its annotation file contains ship category information (only ships here, labeled 0) and normalized bounding box coordinate information. 80% of the dataset is used to train models, and 20% is used for testing. Figure 5 shows some samples of the training dataset.

4.2. Experiment. The LSDM and LSDM-tiny are implemented by PyTorch, and their performances are investigated and compared with YOLOv3 and YOLOv3-tiny on the abovementioned dataset under an NVIDIA GTX1060 (3 g) environment. The evaluation indicators include precision, AP (Average Precision), recall, F1, parameter number, and FPS (Frames Per Second). The value of FPS is the average

FIGURE 6: Detection effect comparison: (a) single ship detection; (b) multiple ships detection.

number of detected images per second obtained by detecting 1000 images through the detection network.

The experiment results are shown in Table 5:

Figure 6 shows the detection effect of the four models, LSDM-tiny, YOLOv3-tiny, LSDM, and YOLOv3. Figure 6(a) is the case for single ship detection, and Figure 6(b) is the case for multiple ships detection.

4.3. Analysis. The abovementioned results show that the LSDM has higher FPS than YOLOv3. As expected, with only one-third parameters, the LSDM is faster than YOLOv3 when detecting ships. More importantly, the LSDM also has better performances in “Recall,” “Precision,” “AP,” and “F1” than YOLOv3. That is, the LSDM has higher accuracy than YOLOv3.

It can be clearly observed in the detection effect image for single ship detection in the Figure 6(a) that the LSDM is superior to other models in terms of positioning accuracy and classification effect due to its deeper network depth. For multiple ship detection in the Figure 6(b), it also can be observed that the LSDM’s positioning is the most accurate and has no problems of missed detection and misdetection happened in other models due to its retaining of shallow features. From the comparison results, the effect of feature reusing in densely connected units is validated.

The abovementioned results also show that LSDM-tiny is much faster than the LSDM and YOLOv3 as expected. The “FPS” of LSDM-tiny is about double that of YOLOv3, but a bit less than that of YOLOv3-tiny. However, LSDM-tiny has a higher accuracy than YOLOv3-tiny. It can also be clearly observed in the detection effect image in Figure 6 that the densely connected units of LSDM-tiny make the classification score and positioning accuracy better than YOLOv3-tiny both in single ship and multiple ship detection cases. This comparison results further validate that it is important for improving detection accuracy of a small object to keeping shallow features into the final global features.

5. Conclusions

This paper proposes a lightweight ship detection model (LSDM) based on YOLOv3 and DenseNet. In the LSDM, the features of the shallow layer are allowed to be retained and used in subsequent layers. This mechanism reduces, greatly, the parameters and optimizes the structure of the backbone network, and spatially separated convolution is used to further reduce the parameters in the feature pyramid network. The two improvements make the parameters of the LSDM be only one-third of YOLOv3. Also, the experimental results show that the LSDM is not only faster than YOLOv3 but also has higher accuracy.

Furthermore, a model called LSDM-tiny is constructed as a simple version of the LSDM. By reducing the number of densely connected units, the parameters of LSDM-tiny is only one-eighth of YOLOv3. The experimental results show that the detection speed of LSDM-tiny is about double that of YOLOv3, with losing a little accuracy. Also, comparing with YOLOv3-tiny, LSDM-tiny has a similar detection speed

but has a higher accuracy due to the reuse mechanism of feature maps.

The LSDM and LSDM-tiny are proposed for fast ship detection on existing normal even poor hardware condition. In the future, two aspects will be studied further. First, for the problem of uneven detection of positive and negative samples in YOLOv3, how to add a stricter penalty mechanism to reduce the impact of negative samples will be studied. Secondly, to detect a small ship object in the camera images, how to increase multiscale detection channels while maintaining a small number of parameters will be studied.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by “the Fundamental Research Funds for the Central Universities” under Grant 3132019400.

References

- [1] X. Chen, L. Qi, Y. Yang et al., “Video-based detection in infrastructure enhancement for automated ship recognition and behavior analysis,” *Journal of Advanced Transportation*, vol. 2020, Article ID 7194342, 12 pages, 2020.
- [2] X. Chen, H. Chen, H. Wu et al., “Robust visual ship tracking with an ensemble framework via multi-view learning and wavelet filter,” *Sensors*, vol. 20, no. 3, p. 932, 2020.
- [3] I. Im, D. Shin, and J. Jeong, “Components for smart autonomous ship architecture based on intelligent information technology,” *Procedia Computer Science*, vol. 134, pp. 91–98, 2018.
- [4] J.-Y. Zhuang, L. Zhang, S.-Q. Zhao, J. Cao, B. Wang, and H.-B. Sun, “Radar-based collision avoidance for unmanned surface vehicles,” *China Ocean Engineering*, vol. 30, no. 6, pp. 867–883, 2016.
- [5] R. Girshick, “Fast R-CNN,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, December 2015.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [7] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, “Mask R-CNN,” in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Venice, Italy, October 2017.
- [8] W. Liu, D. Anguelov, D. Erhan et al., “Single shot MultiBox detector,” in *Proceedings of the Computer Vision—ECCV 2016*, pp. 21–37, Amsterdam, The Netherlands, October 2016.
- [9] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and*

- Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2015.
- [10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.
 - [11] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, “AlexNet-level accuracy with 50x fewer parameters and <1MB model size,” *ArXiv* 2017, abs/1602.07360, 2017.
 - [12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, Honolulu, HI, USA, July 2017.
 - [13] A. G. Howard, M. Zhu, B. Chen et al., “Mobile nets: efficient convolutional neural networks for mobile vision applications,” *ArXiv* 2017, abs/1704.04861, 2017.
 - [14] X. Zhang, X. Zhou, M. X. Lin, and J. Sun, “ShuffleNet: an extremely efficient convolutional neural network for mobile devices,” in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)*, New York, NY, USA, June 2018.
 - [15] J. Redmon and A. Farhadi, “YOLOv3: an incremental improvement,” *ArXiv* 2018, abs/1804.02767, 2018.
 - [16] W. Fang, F. Zhang, V. Sheng, and Y. Ding, “A method for improving CNN-based image recognition using DCGAN,” *Computers, Materials & Continua*, vol. 57, no. 1, pp. 167–178, 2018.
 - [17] R. H. Meng, S. G. Rice, J. Wang, and X. Sun, “A fusion steganographic algorithm based on faster R-CNN,” *CMC-Computers Materials & Continua*, vol. 55, no. 1, pp. 1–16, 2018.
 - [18] Q. Cui, S. McIntosh, and H. Y. Sun, “Identifying materials of photographic images and photorealistic computer generated graphics based on deep CNNs,” *CMC-Computers Materials & Continua*, vol. 55, pp. 229–241, 2018.
 - [19] Z. Dong and B. Lin, “Learning a robust CNN-based rotation insensitive model for ship detection in VHR remote sensing images,” *International Journal of Remote Sensing*, vol. 41, no. 9, pp. 3614–3626, 2020.
 - [20] X. Yang, H. Sun, X. Sun, M. Yan, Z. Guo, and K. Fu, “Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network,” *IEEE Access*, vol. 6, pp. 50839–50849, 2018.
 - [21] W. Fan, F. Zhou, X. Bai, M. Tao, and T. Tian, “Ship detection using deep convolutional neural networks for PolSAR images,” *Remote Sensing*, vol. 11, no. 23, p. 2862, 2019.
 - [22] S. Zhang, R. Wu, K. Xu, J. Wang, and W. Sun, “R-CNN-Based ship detection from high resolution remote sensing imagery,” *Remote Sensing*, vol. 11, no. 6, p. 631, 2019.
 - [23] J. Jiao, Y. Zhang, H. Sun et al., “A densely connected end-to-end neural network for multiscale and multiscene SAR ship detection,” *IEEE Access*, vol. 6, pp. 20881–20892, 2018.
 - [24] K. Kim, S. Hong, B. Choi, and E. Kim, “Probabilistic ship detection and classification using deep learning,” *Applied Sciences*, vol. 8, no. 6, p. 936, 2018.
 - [25] Q. An, Z. Pan, L. Liu, and H. You, “DRBox-v2: an improved detector with rotatable boxes for target detection in SAR images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 8333–8349, 2019.
 - [26] B. Liu, S. Wang, J. Zhao, and M. Li, “Ship tracking and recognition based on Darknet network and YOLOv3 algorithm,” *Journal of Computer Applications*, vol. 39, pp. 1663–1668, 2019.
 - [27] Y. Zhang, J. Shu, L. Hu, Q. Zhou, and Z. Du, “A ship target tracking algorithm based on deep learning and multiple features,” *Society of Photographic Instrumentation Engineers*, vol. 11433, 2020.
 - [28] Y. Wang, C. Wang, and H. Zhang, “Combining a single shot multibox detector with transfer learning for ship detection using sentinel-1 SAR images,” *Remote Sensing Letters*, vol. 9, no. 8, pp. 780–788, 2018.
 - [29] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, “Automatic ship detection based on RetinaNet using multi-resolution gaofen-3 imagery,” *Remote Sensing*, vol. 11, no. 5, p. 531, 2019.
 - [30] Y.-L. Chang, A. Anagaw, L. Chang, Y. Wang, C.-Y. Hsiao, and W.-H. Lee, “Ship detection based on YOLOv2 for SAR imagery,” *Remote Sensing*, vol. 11, no. 7, p. 786, 2019.
 - [31] T. Zhang and X. Zhang, “High-speed ship detection in SAR images based on a grid convolutional neural network,” *Remote Sensing*, vol. 11, no. 10, p. 1206, 2019.
 - [32] L. Qi, B. Y. Li, L. K. Chen et al., “Ship target detection algorithm based on improved faster R-CNN,” *Electronics*, vol. 8, no. 9, p. 959, 2019.
 - [33] Z. Lin, K. Ji, X. Leng, and G. Kuang, “Squeeze and excitation rank faster R-CNN for ship detection in SAR images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 5, pp. 751–755, 2019.
 - [34] Z. X. Zhang, H. L. Li, G. Q. Zhang et al., “A high-speed cascaded convolutional neural network for ship detection with multispectral images,” *Journal of Infrared and Millimeter Waves*, vol. 38, no. 3, pp. 290–295, 2019.
 - [35] T. W. Zhang, X. L. Zhang, J. Shi, and S. J. Wei, “Depthwise separable convolution neural network for high-speed SAR ship detection,” *Remote Sensing*, vol. 11, no. 21, p. 2483, 2019.
 - [36] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.